APPLICATION FOR

UNITED STATES LETTERS PATENT

SPECIFICATION

Inventor(s): Hiroyuki HIGUCHI

Title of the Invention: MULTI-CYCLE PATH ANALYZING METHOD

MULTI-CYCLE PATH ANALYZING METHOD

Background of the Invention

Field of the Invention

5   The present invention relates to a multi-cycle path analyzing method for detecting a multi-cycle path which can take 2 cycles or more for a signal to propagate from the starting point of the path to its end point, for example, among paths between two memory elements within

10   a circuit to be analyzed in a timing analysis and logic verification of a digital circuit.

Description of the Related Art

      A multi-cycle path in a digital circuit is a path

15   that can take time of two cycles or more for a signal propagate from a memory element at a starting point to that at an end point, for example, between the two memory elements. Generally, a path between memory elements is handled as a single cycle path on which a signal propagates

20   in one cycle.

      If a delay between the memory elements becomes large, a design becomes difficult when the path between the memory elements is handled as a single cycle path. By detecting a multi-cycle path which may take two cycles

25   or more from the starting point of the path to its end

point, restrictions are eased to be able to facilitate a design.

Conventionally, not many techniques that are effective for a multi-cycle path analysis exist, and normally, a human being specifies a multi-cycle path at the time of design.

The following document exists as a conventional technique in the case where a pure logic circuit is targeted.

H. Higuchi: An Implication-based Method to Detect Multi-Cycle Paths in Large Sequential Circuits, IEEE/ACM Design Automation Conf. pp.164-169, 2002

However, in an actual normal digital circuit, logic gates are mapped by cells of a cell library. Therefore, two clocks or more exist, and a gated clock is used. Accordingly, it has been difficult to automatically detect a multi-cycle path in an actual digital circuit.

Additionally, there has been a method calculating the state transition of a circuit, namely, reachability from the initial state by recognizing the entire circuit as one finite state machine, and using the reachability so as to detect a multi-cycle path. This method is described by the following document.

K. Nakamura, et al., "Waiting False Path Analysis of Sequential Logic Circuits for Performance

Optimization", IEEE/ACM ICCAD 98, pp.392-395

However, since an entire circuit is handled as one finite state machine with this method, the scale of a finite state machine becomes enormous if the circuit becomes complex. Therefore, this method is not always effective as a practical method for detecting a multi-cycle path.

Furthermore, with a conventional method detecting a multi-cycle path by altogether recognizing paths between two memory elements, it cannot be determined whether or not a targeted multi-cycle path is actually activated, namely, whether or not a change in the value of a memory element at a starting point propagates to a memory element at an end point over the path, or whether or not a change in the value of a memory element at a point other than the starting point propagates to the memory element at the end point.

With an increase in the scale of a circuit in recent years, a multi-cycle path analysis has become complex, and its processing time has become large. Besides, if a clock is complex such as in the case where a gated clock is used, an influence of the clock cannot be accurately determined, and there are difficulties in the detection of a multi-cycle path.

Additionally, with a conventional method

detecting a multi-cycle path by altogether recognizing paths between two memory elements, it is not examined whether or not a targeted multi-cycle path is actually activated. Therefore, an ability to detect a multi-cycle

5   path is degraded.

Furthermore, if a multi-cycle path is specified by a designer at an early stage of design, the specification of the multi-cycle path is normally fixed, and a path the timing of which is strict is not actually

10 specified at a stage after the design. As a result, a timing restriction is not satisfied. Besides, variations in a circuit delay in chip manufacturing tend to increase. However, since a timing restriction is fixed at the time of design, a product yield is degraded.

15

Summary of the Invention

An object of the present invention is to provide a multi-cycle path analyzing method which can also cope with the use of a gated clock and an increase in the

20 scale of a circuit, has a short processing time, and can accurately detect a multi-cycle path in view of the above described problems.

Another object of the present invention is to provide an analyzing method which improves an ability

25 to detect a multi-cycle path by examining a path

activation condition, and can also cope with variations in a product delay by examining the movability of a multi-cycle path even in the case where a multi-cycle path is specified at the time of design.

5   To achieve the above described objects, a multi-cycle path analyzing method according to the present invention, to detect a multi-cycle path that can take two cycles or more for a signal to propagate from a starting point of the path to its end point among

10   paths within a circuit to be analyzed, comprises: analyzing a circuit to be analyzed in correspondence with the name of each element which includes a cell configuring the circuit to be analyzed, and/or the meaning and/or the relationship of a signal to a terminal of

15   each element; and making a multi-cycle path determination of whether or not a path from a starting point to an end point is a multi-cycle path by using a result of the analysis.

        Additionally, to achieve the above described

20   objects, a multi-cycle path analyzing apparatus according to the present invention, to detect a multi-cycle path that can take two cycles or more for a signal to propagate from a starting point of the path to its end point among paths within a circuit to be analyzed,

25   comprises: a circuit converting unit performing circuit

conversion for making a multi-cycle path analysis in correspondence with the meaning of a signal to a terminal of each element which includes a cell configuring a circuit; and a multi-cycle path determining unit making

5   a determination of whether or not a path from a starting point to an end point is a multi-cycle path by using a result of the circuit conversion.

Furthermore, to achieve the above described objects, a program according to the present invention

10   for causing a computer, which makes an analysis of a multi-cycle path that can take two cycles or more for a signal to propagate from a starting point of a path to its end point among paths within a circuit to be analyzed, to execute a process, the process comprises: performing

15   circuit conversion for a multi-cycle path analysis in correspondence with the meaning of a signal to a terminal of each element which includes a cell configuring a circuit; and making a determination of whether or not a path from a starting point to an end point is a multi-cycle

20   path by using a result of the circuit conversion.

Brief Description of the Drawings

Fig. 1 is a block diagram showing the principle of a multi-cycle path analyzing method according to the

25   present invention;

Fig. 2 is a flowchart showing a fundamental process for detecting a multi-cycle path;

Fig. 3 is a flowchart showing a fundamental process for analyzing the movability of a multi-cycle path;

5   Fig. 4 is a block diagram showing the configuration of a data processing device for implementing a multi-cycle path analyzing method according to a preferred embodiment;

Fig. 5 is a flowchart showing the whole of a process

10   for analyzing a multi-cycle path;

Fig. 6 is a flowchart showing the details of a process for performing conversion of a gated clock into an enable signal to a flip—flop;

Fig. 7 explains a specific example of the process

15   for performing conversion of a gated clock;

Fig. 8 shows a result of the circuit conversion shown in Fig. 7;

Fig. 9 is a flowchart showing the details of a process for performing conversion of a timing condition

20   of an external terminal into a restriction circuit;

Fig. 10 explains a specific example (No. 1) of the conversion into the restriction circuit in Fig. 9;

Fig. 11 explains a specific example (No. 2) of the conversion into the restriction circuit in Fig. 9;

25   Fig. 12 explains a specific example (No. 3) of the

conversion into the restriction circuit in Fig. 9;

Fig. 13 is a flowchart showing the details of a process for counting up a reachable state of a finite state machine;

5   Fig. 14 explains a specific example of the process shown in Fig.13;

Fig. 15 is a flowchart showing the details of a process for converting a selection control input of a selector into an enable signal;

10   Fig. 16 explains a specific example of the conversion process shown in Fig. 15;

Fig. 17 explains a specific example of a change in an enable signal;

Fig. 18 is a flowchart showing the details of a

15   process for determining an activation condition of all of the paths between a starting point and an end point;

Fig. 19 explains a specific example of the process for determining the movability of a multi-cycle path; and

20   Fig. 20 is a flowchart showing the details of the process for determining the movability of a multi-cycle path.

Description of the Preferred Embodiments

25   Fig. 1 is a block diagram showing the principle

of a multi-cycle path analyzing method according to the present invention. This figure is a block diagram showing a method analyzing a multi-cycle path that may take two cycles or more for a signal to propagate from a starting

5   point of a path to its end point among paths within a circuit to be analyzed.

In Fig. 1, in 1, an analysis of a circuit to be analyzed is made in correspondence with the name of each element which includes a cell configuring the circuit,

10   and the meaning and/or the relationship of a signal to a terminal of the element, and in 2, a multi-cycle path determination of whether or not a path from a starting point to an end point is a multi-cycle path is made.

In a preferred embodiment according to the present

15   invention, circuit conversion for a multi-cycle path analysis may be performed in correspondence with the meaning of a signal to a terminal of each element in the circuit analysis 1, and the multi-cycle path determination 2 may be made by using a result of the

20   conversion.

In this case, circuit conversion for converting a signal, which can be converted into an enable signal to memory elements at starting and end points of the path among signals to the terminal of the element, into

25   an enable signal can be performed, and a multi-cycle

path determination can be also made depending on whether or not the value of a memory element can possibly change based on the value of the enable signal to the memory elements at the starting and the end points in the

5   multi-cycle path determination 2.

Additionally, in the circuit conversion, circuit conversion for converting a selection control signal to a selector which controls the settings of values of memory elements at a starting point and an end point

10   into an enable signal, circuit conversion for converting an enable signal to a memory element in a source of a clock into an enable signal to the memory elements at the starting and the end points when the source of the clock which drives the memory elements at the starting

15   and the end points is a memory element, and circuit conversion for converting the driving clock into an enable signal by using clock gating information of the clock which drives the memory elements at the staring and the end points can be also performed.

20   Additionally, in a preferred embodiment, if all of the paths between the starting and end points of the path are inactive for the path which is determined not to be a multi-cycle path in the multi-cycle path determination made based on the value of an enable signal,

25   the path can be also determined to be a multi-cycle path.

Furthermore, in a preferred embodiment, based on a condition for the relationship between a value setting signal to an external input terminal in a circuit, and a value read signal from an external output terminal,

5   a restriction circuit corresponding to the condition can be added to a circuit to be analyzed in the circuit analysis 1, and the multi-cycle path determination 2 can be also made for the circuit to be analyzed after the addition. Besides, memory elements within the circuit

10   can be classified into groups by the names of the respective memory elements, and the reachable state of a finite state machine represented by each of the groups can be calculated in the analysis 1, and the multi-cycle path determination 2 can be made by using a result of

15   the calculation.

Furthermore, in a preferred embodiment, if one memory element or more exist on a path between memory elements at starting and end points of the path among paths determined to be a multi-cycle path, and if a

20   multi-cycle path restriction is imposed in some of partial paths between two successive memory elements at the time of circuit design, it can be further examined whether or not the multi-cycle path restriction can be moved to a different partial path according to an enable signal

25   to the memory elements at the starting and the end points.

Still further, as described above, information required for performing circuit conversion can be stored instead of actually performing circuit conversion in the analysis 1 of a circuit to be analyzed, and the

5   multi-cycle path determination 2 can be made by using the stored contents.

Still further, a multi-cycle path analyzing apparatus according to a preferred embodiment comprises a circuit converting unit performing circuit conversion

10   for the above described multi-cycle path analysis, and a multi-cycle path determining unit determining a multi-cycle path by using a result of the circuit conversion. Besides, the circuit converting unit can also perform circuit conversion for converting a signal

15   which can be converted into an enable signal to memory elements at starting and end points of a path into an enable signal.

Still further, in a preferred embodiment, used as a program for implementing a multi-cycle path analyzing

20   method is a program for causing a computer to execute a process, the process comprising: performing circuit conversion for a multi-cycle path analysis; and determining a multi-cycle path by using a result of the conversion.

25   As described above, according to the present

invention, circuit conversion for a multi-cycle path analysis is performed, for example, in correspondence with the meaning of a signal to a terminal of each element configuring a circuit, and a multi-cycle path

5   determination is made by using a result of the conversion.

Fig. 2 explains a fundamental process for detecting a multi-cycle path in a preferred embodiment according to the present invention. In this figure, firstly, in step S1, data required for a multi-cycle path detection

10   process is input. As this data, a circuit to be detected, paths in the circuit, a pair of a memory element, for example, as a starting point and a memory element at an end point, to which a signal from the memory element at the starting point is input over the path, value

15   settings of external input/output, a restriction on the relationship between times at which data is read from the memory elements at the starting and the end points, and the like are input. The value settings of external input/output, and the restriction on the relationship

20   between the read times will be described later.

Processes from the next step S2 to step S6 are the essential processes for detecting a multi-cycle path. Firstly, in step S2, all of signals that can be converted into, by way of example, an enable signal to a memory

25   element are converted into enable signals among various

types of signals. In this preferred embodiment, multi-cycle path detection is made depending on values stored in the memory elements at the starting and the end points of a path within the circuit, for example,

5 a change in the values of flip-flops. However, whether or not a change occurs in the values of flip-flops is fundamentally controlled according to the value of an enable signal. Therefore, multi-cycle path detection is made based on the value of an enable signal.

10 In step S2, a selection control input to a value setting control selector is first converted into an enable signal as a signal which can be converted into such an enable signal. The value setting control selector is a selector for controlling the settings of the values

15 in flip-flops, for example, at the starting and end points as will be described later, and the selection control input to the selector is converted into an enable signal.

A second signal converted into an enable signal is a gated clock. A gated clock is given in the form

20 where, for example, an AND gate is inserted between an original clock signal and a clock terminal of a flip-flop, and an output of the AND gate is determined based on a condition of an input to the AND gate.

For the third signal conversion, in the case where

25 a memory element is further connected, for example, to

a clock terminal of a flip-flop, and the memory element is used as a clock source, an enable signal to the memory element is converted into an enable input signal to a flip-flop at a starting or an end point.

5   In step S3, enable input signals to the memory elements at the starting and the end points including the signals converted in step S2 are extracted. Then, in step S4, circuits equivalent to the external input/output value settings described in step S1, and

10   a restriction on the relationship between the read times at the starting and the end points are added. Also this process will be described later.

Then, in step S5, a local finite state machine (FSM) within the circuit is identified, and a detection process

15   for counting up a reachable state from its initial state is executed. Also its details will be described later.

In step S6, it is examined whether or not the flip-flop at the end point is enabled, for example, at a time next to the time when the flip-flop at the starting

20   point is enabled. If the flip-flop at the end point is enabled at the next time, it means that the signal reaches from the starting point to the end point in one cycle. Therefore, that path is not a multi-cycle path but a single-cycle path.

25   Additionally, in step S6, it is examined whether

or not the path from the starting point to the end point is activated, namely, whether or not the signal from the starting point reaches the end point in one cycle. If all of the paths between the starting and the end

5 points are not activated, it means that the path connecting the starting and the end points is a multi-cycle path.

With the above described processes, the fundamental process for detecting a multi-cycle path

10 is terminated, and a result of the multi-cycle path determination is output in step S7. Here, the process for detecting a multi-cycle path is completed.

Fig. 3 is a flowchart showing a fundamental process for detecting the movability of a multi-cycle path, which

15 examines whether or not a multi-cycle path restriction specified, for example, at the time of design can be moved on a multi-cycle path by examining whether or not a different memory element exists on the path between a memory element at a starting point and a memory element

20 at an end point, namely, whether or not the path straddles a memory element, and by targeting the multi-cycle path that straddles such a memory element.

Firstly, in step S8, a circuit to be targeted is specified.

25 In step S9, a path which straddles a memory element

is recognized as a detection target, and a multi-cycle path which exceeds the number of cycles (the number of straddled memory elements +1) on a path between a starting point and an end point is detected. For example, if one

5 flip-flop exists on a path between a flip-flop at the starting point and a flip-flop at the end point, (the number of straddled memory elements +1) results in 2, and a multi-cycle path which exceeds two cycles, namely, a multi-cycle path which takes 3 cycles or more is

10 detected.

Then, in step S10, movability such that the multi-cycle path can move to a path between which memory elements between the starting and the end points is output. If a multi-cycle path which straddles one memory element

15 on the path and takes three cycles or more is detected as described above, a multi-cycle path restriction can be moved between the memory element on the path and the end point when a specified path is, for example, between the starting point and the memory element on the path

20 in the case where the multi-cycle path specified, for example, at the time of design is either between the starting point and the memory element on the path or between the memory element on the path and the end point, and this movability is output in step S10.

25 Fig. 4 is a block diagram showing the configuration

of a data processing device when the fundamental processes

shown in Figs. 2 and 3 are executed by using the data

processing device, namely, a computer. In this figure,

an input/output device 11 and a storing device 12 are

5   connected to the data processing device 10. The

input/output device 11 is intended to read circuit data,

to input a user operation, to output multi-cycle path

data resultant from detection, and the like. The storing

device 12 is intended to store circuit data, an

10   intermediate result of a process, and the like.

        The data processing device 10 comprises a

starting/end point instructing unit 13 instructing a

pair of starting and end points of a path for which a

multi-cycle path analysis is to be made in a given circuit,

15   an enable signal converting unit 14 executing the process

in step S2 of Fig. 2, an enable input extracting unit

15 executing the process in step S3, an external terminal

value setting/read time restriction circuit generating

unit 16 executing the process in step S4, a local FSM

20   reachability analyzing unit 17 executing the process

in step S5, a multi-cycle path analyzing unit 18 executing

the process in step S6, and a memory-element-straddling

multi-cycle path analyzing unit 19 executing the process

in step S9 of Fig. 3.

25   Fig. 5 is a flowchart showing the details of the

process for analyzing a multi-cycle path in this preferred

embodiment. The process shown in this figure is described

in detail with reference to Figs. 6 to 20 showing the

details of a process in each step and specific examples

5   of circuits.

Once the process is started in Fig. 5, a circuit

C for which the analysis process is to be executed, a

set P of pairs of starting and end points, and a timing

condition E of an external terminal are first input in

10   step S12. Examples of the timing condition E include

a condition where the value of an external input B does

not change during 5 clocks from when the value of an

external input A changes, a condition where the value

of an external output D is observed only after 3 clocks

15   from when the value of A changes, and the like.

In step S13, a gated clock of a clock which drives

each memory element in a portion (such as a logic operation

unit only) other than a clock tree within the circuit

C, by way of example, a flip—flop FFi, and an enable

20   input to a memory element (within the clock tree), which

is a clock source, are converted into enable signals

of the FFi.

In this process, it is examined whether or not the

clock which drives, for example, the flip-flop FFi in

25   a portion other than the clock tree is a gated clock

which masks a signal from the clock source according to a condition, and a circuit corresponding to a condition where the signal from the clock source is not masked is synthesized and converted into an enable signal ENgated

5 of the FFi.

Additionally, if the clock source is a memory element, an enable input to the memory element is defined as an enable signal ENclksrc of the FFi unchanged. If the original enable input to the FFi is defined to be

10 ENorig, an AND gate which uses these three enable signals as fan-in is generated, and its output is connected to an enable input of the FFi.

Fig. 6 is a flowchart showing the details of the process in step S13. The process shown in this figure

15 is described with reference to the specific examples of circuits shown in Figs. 7 and 8.

Once the process is started in Fig. 6, information of a clock signal, in Fig. 7 information indicating that a signal x from a clock terminal is a clock signal is

20 input. In step S32, an analysis of a clock tree CT is made.

With this analysis, fan-out is traced until a cell or a module other than a logic gate, here, an FF21 is reached in the output direction from a terminal which

25 outputs the clock signal in Fig. 7, namely, in the

direction to a clock input terminal of the FF, and the partial circuit traced at that time is recognized as a clock tree CT.

In Fig. 7, the partial circuit, which is configured
5   by the path which transmits the signal x, an AND gate 22, and a path which connects an output f of the AND gate 22 to the clock terminal of the FF21, is a clock tree. The other input to the AND gate, namely, a path which transmits g1 is not included in the clock tree
10 CT.

In step S33, a logic variable g1 is assigned to the root of the clock tree CT, the clock terminal in Fig. 7, and an output terminal of a node other than the CT, namely, a terminal of a partial circuit 23 which
15 outputs g1, and a calculation of a logic function in a leaf of the CT, namely, the clock input terminal to the FF is made.

In Fig. 7, only one FF exists, and the number of leaves of the CT is only one. Generally, however, a
20 plurality of FFs exist within a circuit C. Therefore, the calculation of the logic function in the clock input terminal is made for all of FFs in step S33. The logic function calculated here becomes the following equation for Fig. 7.

25      $f = x \cdot g1$

Then, in step S34, a set A of FFs whose clock input terminals are connected to the leaves of the clock tree CT is obtained. In Fig. 7, this set A includes only one FF21 as an element.

5   In step S35, it is determined whether or not the set A is empty. If the set A is not empty, one FF is extracted from the set A in step S36, and defined as FFi. Then, in step S37, executed is a process for determining the clock signal as a positive phase clock

10   if the logic function f in the leaf of the clock tree CT depends on the signal x from the clock terminal as the logic variable at the root of the tree, namely, if x is included unchanged, or for determining the clock signal as a reverse phase clock if the logic function

15   f depends on x', namely, an inversion signal of x, that is, if x' is included. In step S38, a logic function F which represents a condition where a gated clock is enabled is obtained in correspondence with a result of the determination.

20   That is, in step S38, the logic function F is obtained with the following equation in correspondence with the positive phase clock and the reverse phase clock by using the value of f when x within the logic function f is set to 1, or a value obtained by inverting the value

25   of f when x is set to 0. In the following equation, *

indicates multiplication.

In case of the positive phase clock

$$F=f|x \leftarrow 1 * (f|x \leftarrow 0)'$$

In case of the reverse phase clock

5           $$F=f|x \leftarrow 0 * (f|x \leftarrow 1)'$$

In Fig. 7, since the logic function f depends on x, it is determined to be the positive phase clock in step S37. In step S38, the value of f when 1 is substituted for x in the equation for the positive phase clock becomes

10   g1, whereas the value of f when 0 is substituted for x, namely, the inversion value of 0 becomes 1. Therefore, the value of F results in g1.

In step S39 of Fig. 6, a logic circuit calculating this logic function F is generated, and an input of the

15   circuit is connected to a corresponding terminal within the circuit C, and an output of the generated circuit is defined as an enable signal ENgated to the FFi. Then, the processes in and after step S35 are repeated, and the process is terminated when the set A is determined

20   to be empty in step S35.

Fig. 8 shows the circuit generated in step S39 in correspondence with Fig. 7. In this circuit, the output of the circuit generating F, namely, g1 is defined as ENgated unchanged and input to an AND gate 25 along with

25   the original enable signal ENorig, and an output of the

AND gate 25 is input to an enable terminal of the FF21. In the meantime, the signal x from the clock terminal is input to the clock terminal unchanged.

The process goes back to Fig. 5. Then, in step S14, the timing condition E is converted into a restriction circuit for the external terminal, and added to the circuit C. By way of example, for the condition where the value of the external input B does not change during 5 clocks from when the value of the external input A changes, a counter which is reset to 0 at a time next to the time when the value of the external input A changes, increments its value up to 5, and holds that value thereafter is generated, and a circuit which externally stores a value in the external input B only when the value of the counter is 5 is added.

Fig. 9 is a flowchart showing the details of the process in step S14. Once the process is started in this figure, an external input value setting/read time restriction is first extracted, for example, from a memory in step S41. This restriction corresponds to the timing condition E of the external terminal, which is input in step S12 of Fig. 5.

Then, in step S42, an additional circuit shown in Fig. 10 is configured for the value setting restriction for an external input, namely, a restriction such that

a value is not set for an input PI2 for a time K from when the value of an input PI1 is set. This restriction corresponds to a time delay in the giving of corresponding data to the PI2, for example, if an address is externally

5   specified as the PI1.

In Fig. 10, a circuit using an FF31 and an XOR 32, and a circuit using an FF33 and a selector 34 are respectively added to external input terminals PI1 and PI2.

10   In Fig. 10, if the value of an external input signal PI2 changes, for example, from 0 to 1, this value is given to the external input terminal PI1, and also input to a data input to the FF31 and either of inputs of the XOR 32. At this time, the output of the FF31 is the former

15   value of the PI1, namely, 0, and the output X of the XOR 32 becomes 1 at the time point when the PI1 changes from 0 to 1, for example, at a time t.

Therefore, a relationship which gives, as the following equation, a selection control signal C to the

20   selector 34 connected to the input side of the external input terminal PI2 is stored, for example, in a memory in correspondence with $X(t)=1$, and used for a calculation at and after the time t.

$$X(t)=1 \rightarrow C(t+1)=0$$

25   $$C(t+2)=0$$

$$:$$

$$C(t+K)=0$$

Such a stored relationship is used, whereby the value of the selection control signal C to the selector 34 becomes 0 from a time t+1 until a time t+K regardless of the value of the external input PI2. Additionally, the selector 34 selects the data input 0, namely, the output of the FF33, whereby the value of the external input PI2 is not set from the time t+1 until the time t+K.

The process goes back to Fig. 9. In step S43, an additional circuit is configured for the next external input value setting restriction such that "the value of the PI2 is not set if PI1=v" in step S43. Here, v may be either 0 or 1. An example of the additional circuit when v is 0 is shown in Fig. 11.

In Fig. 11, a circuit configured by a selector 35 and a flip-flop 36 is added before the input terminal PI2. If the external input PI1 is 0, the selector 35 selects the output of the FF36. At this time, the formerly set value of the PI2, for example, 0 is stored in the FF36, and the formerly set value, for example, 0 is always output as the output of the FF36 to the external input terminal PI2. Even if a new value is attempted to be externally set as the PI2, that value is not set as long

as the PI1 is 0.

Fig. 12 shows an example of the additional circuit in the case where v is set to, for example, 0 in the external output value read restriction such that the value of the PO2 is not read out if the PO1 is v, again in step S44 of Fig. 9.

In Fig. 12, a circuit configured by a selector 37 and an FF38 is added to the output side of an external output terminal PO2. If the external output PO1 is 0, the selector 37 outputs, for example, 0 as the value of the FF38, namely, the formerly read value of the PO2. Even if the value of the PO2 is attempted to be externally read out, a new value cannot be read out. The new value of the PO2 cannot be externally read out until the PO1 changes to 1.

Then, the memory elements within the circuit C are classified into groups in step S15 of Fig. 5, and a reachable state of a finite state machine (FSM), which is represented by each of the groups, is counted up. In this process, the memory elements within the circuit C are classified into groups, for example, by names, and all of the states reachable from the initial state are counted up by recognizing respective groups as local FSMs, and that information is held.

Fig. 13 is a flowchart showing the details of this

process in step S15. This process is described with reference to Fig. 14.

Once the process is started in Fig. 13, a group having a clear terminal, or a group having a preset terminal is recognized as one FSM among groups which are externally specified or automatically obtained, for example, according to a similarity of names assigned to respective flip-flops.

By way of example, two FF groups both of which have a clear terminal and similar names such as FSM-REG-1 and FSM-REG-0 are recognized as one FMS. Since both of these groups have the clear terminal here, the initial state of this FMS becomes as follows by setting the clear input to 1.

$$(FSM\text{-}REG\text{-}1, FSM\text{-}REG\text{-}0) = (0, \ 0)$$

A process for obtaining this initial state is step S47. In step S47, the initial state of an FSM is obtained depending on whether each FFs of the FSM have either a clear terminal or a preset terminal. Namely, the initial value of FFs having the clear terminal is 0, whereas that of FFs having the preset terminal is 1.

Then, in step S48, all of states reachable from the initial state are obtained. Fig. 14 explains a state transition as a result of the state search. In this figure, states reachable from the initial state (0, 0) become

as follows according to the configuration of the circuit
C. It is proved that the transition cannot be made to
a state (1, 0) according to the configuration state of
the circuit C.

5   {00, 01, 11}

Then, an implication relationship is extracted
from a set of the reachable states in step S49. The
implication relationship indicates a unique
relationship among the states extracted from a result
10   of the state search. The following relationship is
extracted as an example.

FSM-REG-1(t)=1 → FSM-REG-1(t)=1

The process goes back to Fig. 5. In step S16, it
is determined whether or not a set P of pairs of starting
15   and end points is empty. Here, the set is not normally
empty, and one pair of a starting point S and an end
point T is extracted from the set P in step S17. A selection
control input of the value setting control selector,
which corresponds to the starting point S and the end
20   point T, is converted into an enable signal for flip-flops
at the starting point S and the end point T.

In this process, for example, if the selector is
directly connected to a D input of the flip-flop at the
starting point S, and if one of data inputs to the selector
25   is equivalent to a Q output of the flip-flop, that selector

is recognized as the value setting control selector for the starting point S, and an output obtained by performing an AND operation between the selection control input of the selector and the enable signal to the flip-flop

5   is reconnected to the enable input of the flip-flop. A similar process is executed also for the flip-flop at the end point T.

Fig. 15 is a flowchart showing the details of the conversion process in step S18. In this figure, the

10   process for the end point T is again executed, for example, after the entire process for the starting point S is completed.

Once the process is started, it is determined in step S51 whether or not the D input to the flip-flop

15   at the starting point is the output of the selector. If the D input is not the output of the selector, the process is immediately terminated. Namely, if the D input is not the output of the selector, the D input to the flip-flop is given, for example, from an AND gate or

20   an OR gate other than the selector, and there is no need to convert a control input of the selector. Therefore, the process is immediately terminated.

If the D input is the output of the selector, it is determined in step S52 whether or not one of the data

25   inputs to the selector is the Q output of the flip-flop.

Also if the D input is not the Q output, it is determined that there is no need to convert the selection control input into an enable signal for that selector, and the process is immediately terminated.

5   If the data input is the Q output, a signal from a circuit which generates a selection control input signal value for selecting the Q output of the flip-flop is converted into an enable signal to the flip-flop at the starting point S as the input data to the selector in 10   step S53, and the process is terminated.

Fig. 16 explains the process in step S53. In this figure, an output of a selector 52 is given, for example, to a D input of a flip-flop 51 at the starting point S, and a Q output of the FF51 is connected to a data 15   input terminal 0 of the selector 52. In this circuit, the output of the FF51 is selected and given to the D input of the FF51 if the selection control signal C to the selector 52 is 0. In the meantime, a signal X corresponding to an external input is given to the D 20   input if the signal C becomes 1. Accordingly, this signal C plays a role of an enable signal to the FF if the selector 52 does not exist, and the external signal X is directly given to the D input of the FF51, so that the signal C can be connected to an enable terminal of the FF 25   unchanged.

As described above, after the process shown in Fig. 15 is terminated for the starting point S, this process is executed for the end point T, and a selection control signal of the selector for the flip-flop at the end point

5   T is converted into an enable signal.

The process again goes back to step S19 of Fig. 5, and a multi-cycle path determination is started. Firstly, in step S19, it is determined whether or not the memory element at the end point T is disabled at

10   a time next to the time when the memory element at the starting point S is enabled.

Namely, if the memory element at the end point T is enabled and its value can possibly change at a time next to the time when the memory element at the starting

15   point S is enabled and its value can possibly change, there is a possibility that the value stored in the memory element at the starting point S needs to reach the memory element at the end point T in one cycle. Therefore, it cannot be said that the path between S and T is a multi-cycle

20   path.

If there is no such possibility, and if the memory element at the end point T is disabled at the next time, the value stored in the memory element at the starting point S does not need to reach the memory element at

25   the end point T in one cycle. Therefore, that path is

proved to be a multi-cycle path.

Fig. 17 explains a specific example of such a multi-cycle path. In this figure, FF1 and FF2 are respectively assumed to be memory elements at the starting point S and the end point T. It is determined whether or not a path in a combinatorial circuit part between the FF1 and the FF2 is a multi-cycle path. If an enable signal to the FF1 is 1 at a time t, both of the outputs of FF3 and FF4 are 0. The outputs of the FF3 and the FF4 respectively become 0 and 1 at the next time t+1, and an enable signal to the FF2 therefore becomes 0. Accordingly, the determination condition in step S19 is proved to be satisfied.

The process goes back to Fig. 5. If the end point T is disabled in step S19, that path is immediately determined to be a multi-cycle path in step S21. Then, the processes in and after step S16 are repeated.

In the meantime, if it cannot be definitely determined that the memory element at the end point T is disabled at the next time in step S19, whether or not the path is a multi-cycle path is determined by examining a path activation condition in step S20.

Namely, in this process, whether or not all of the paths from the starting point S to the end point T are inactive at the time when the memory element at the staring

point is enabled, namely, whether or not a change in the value at the starting point S propagates to the end point T is examined by using a path activation condition to be described later. If all of the paths are determined

5   to be inactive, that path is determined to be a multi-cycle path in step S21, and the processes in and after step S16 are repeated. If all of the paths are determined not to be inactive, and at least one of them is active, the process immediately goes back to step S16.

10   Fig. 18 is a flowchart showing the details of the determination process in step S20. Once the process is started in this figure, it is determined in step S56 whether or not a path yet to be processed between the starting point S and the end point T exists. If such

15   paths exist, one of them is selected in step S57, and examined by using, for example, a static cosensitization condition as an activation condition in step S58. This condition is referred to by the following document.

         S. Davadas, et al., Logic Synthesis, P.239,

20   McGraw-Hill, 1994

         The static cosensitization condition is briefly described. For example, if an AND gate exists on a path, a condition such that a controlled value as its output is 0, and the value of an input on the path among inputs

25   to the AND gate must be 0 as a controlling value is given.

Namely, a signal from a different path is given as a
different input to the AND gate. However, if the value
of an input on the path is 0, the output becomes 0
irrespective of the value of an input from a different
5   path, and an activation condition is satisfied for the
AND gate on that path.

For example, if the controlled value for an OR gate
is 1, and the value of an input on a path is 1 as a controlling
value, an activation condition is satisfied for the OR
10 gate.

If an activation condition is satisfied for all
of the nodes (gates and cells) on the path, the activation
condition is determined to be satisfied in step S59,
and the process is terminated. However, if the activation
15 condition is not satisfied, the processes in and after
step S56 are repeated for a path yet to be processed,
and the process is terminated when no path yet to be
processed is determined to exist.

If the set P of pairs of starting and end points
20 is determined to be empty in step S16 of Fig. 5, a
multi-cycle path on which a memory element exists between
the memory elements at the starting and the end points,
namely, a multi-cycle path which straddles the memory
element is targeted, and a multi-cycle path whose cycle
25 number exceeds the number of cycles (the number of

straddled memory elements +1) is detected in step S22.

If such a multi-cycle path is detected, movability indicating whether or not a multi-cycle path restriction, which is specified, by way of example, for one partial

5 path at the time of design, can move to a different partial path among respective partial paths between the memory elements on the multi-cycle path is output in step S23, and the process is terminated.

Fig. 19 explains a specific example of a multi-cycle

10 path on which the movability of a multi-cycle path restriction is examined. This figure shows part of the circuit C, and assumes that FFA holds a value until 3 clocks elapse from when the value is set, FFB stores a value input from the FFA via a combinatorial circuit

15 in each clock, and FFC stores a value input from the FFB via a combinatorial circuit after 3 clocks elapse from when the value is set in the FFA.

Therefore, a path between the FFA and the FFC, which straddles the FFB, is a multi-cycle path of 3 cycles.

20 However, since only one FFB exists on this multi-cycle path, either the path between the FFA and the FFB or the path between the FFB and the FFC can be set as a 2-cycle path. For example, if a restriction which defines the path between the FFA and the FFB as a multi-cycle

25 path of 2 cycles is input at the time of design, this

multi-cycle path can be considered to be moved between the FFB and the FFC.

Fig. 20 is a flowchart showing the details of step S22 of Fig. 5, namely, the process for detecting a multi-cycle path that exceeds the number of cycles (the number of straddled memory elements +1). In this figure, processes in steps S61 to S65 are executed for all of combinations of FFS within the circuit C, namely, the FF pairs.

Once the process is started, it is determined in step S61 whether or not an FF pair yet to be processed exists. If FF pairs yet to be processed exist, one of them is selected, and its starting and end points are respectively defined as S and T in step S62. In step S63, it is determined whether or not an FF exists between the starting and the end points, namely, whether or not a path between the FFs of the selected pair straddles a memory element. If the path does not straddle a memory element, the processes in and after step S61 are repeated.

If the path straddles a memory element, the numbers of FFs respectively existing on paths which exist in a parallel manner between the starting point S and the end point T are examined, and their maximum number is defined as K in step S64. In step S65, it is examined whether or not a path which straddles the FFs the number

of which is the maximum number K is a multi-cycle path whose cycle number is (K+2) or more. If the path is a multi-cycle path whose cycle number is (K+2) or more, this path is determined to be a multi-cycle path that

5   has movability. Then, the processes in and after step S61 are repeated. The process is terminated when no FF pair yet to be processed is determined to exist in step S61.

In the process for examining whether or not a path

10   is a multi-cycle path of (K+2) cycles or more, the processes in steps S19 and S20 of Fig. 5 can be executed by recognizing that a straddled FF is removed from a path.

By using the above described information of the

15   movability of a multi-cycle path, the multi-cycle path can be moved according to variations in a product delay after a circuit is manufactured, and a timing restriction can be also changed for each product. Additionally, only step S22 of Fig. 5 is applied to a path which is already

20   proved to be a multi-cycle path, and only its movability can be examined.

The above provided description refers to the detection of a multi-cycle path whose cycle number is two or more by assuming that the fundamental number of

25   cycles is 2. However, an analysis of a multi-cycle path

whose cycle number is K can be normally made by examining whether or not a memory element at an end point is enabled in a range from a time next to a time when the value of a memory element at a starting point of the path can

5   possibly change, namely, the time when the memory element at the starting point is enabled, until a (K-1) time from a current time, in a similar manner as in the description provided for step S65 of Fig. 20.

Additionally, the above provided description

10   refers to the preferred embodiment in the form where, for example, a gated clock and an enable input of a memory element of a clock source are converted into signals to an actual enable terminal of a flip-flop as a circuit. However, also considered is a preferred embodiment where

15   such information is held, for example, in a memory, and used for the determinations by referencing this information in steps S19 and S20 of Fig. 5.

Such information may be held in a memory for the conversion of a condition for an external terminal into

20   a restriction circuit in step S14 of Fig. 5, or for the conversion of a selection control input of a selector into an enable signal in step S18, and the determinations can be made by referencing this information in steps S19 and S20. Furthermore, for a large-scale circuit,

25   troublesomeness of examining all of paths in step S20

becomes very high. Therefore, the process in step S20 can be omitted.

As described above, according to the present invention, a change in values of memory elements at starting and end points of a path are determined according to a value of an enable signal. This eliminates the need for extracting a condition where the value changes by analyzing the logic of a circuit. Additionally, by using information of a cell within a circuit, such as a selection control input of a selector, and information of reachability of a local finite state machine, an analysis of a multi-cycle path can be made faster, and an analysis ability can be improved.

Furthermore, since a timing restriction can be satisfied more easily by analyzing the movability of a multi-cycle path without needing the troublesomeness of a designer. As a result, a design time can be reduced. Still further, a result of a movability analysis is applied to an operation for changing a timing restriction according to variations in a delay of a circuit after being manufactured, thereby improving a yield of the product. This greatly contributes to improvements in practicability of a multi-cycle path analysis.